多重世界論理に基づく知識表現

織田　充・田中　譲
（北海道大学・工学部）

1、序論

知識をコンピュータ上で扱うためには、知識表現をどのようにするかが重要な問題になる。知識表現は問題毎に ad hoc に対処するのではなく、それらを統一的に、かつ形式的に取り扱えることが必要である。

本研究においては多重世界モデルを用いて知識を表現することを考える。多重世界を用いる理由は、単一世界では知識の局在化、いいかえればモジュール化を行なうことができないからである。ここでは特定の概念や対象に関する知識に対応する個々の世界が、有向リンクにより関連づけられたヒープ状の構造を考える。この多重世界上に存在する論理構造に注目し、証明可能性を示す様相演算子や、renaming を行なうための演算子、知識の継承方法を規定するいくつかの世界間リンク等を導入する。これらにより従来個々に論じられてきたデフォールトリーズニング[1]、[2]，サーカムスクリプション[3]，閉世界仮説[4]などの非単調論理を用いた知識表現手法を統一的に扱うことができる新しい表現モデルを提案する。

2、多重世界モデル

多重世界は基本的には Fig1 のように丸で表わされた単位世界と矢印により表わされた単位世界を関連づける世界間リンクにより構成される。

任意の世界間リンクについて、矢印の始点にある世界を親世界、終点にある世界を子世界と呼ぶ。

2、1 単位世界

単位世界は基本的には多重世界論理における論理式の集合である。それぞれの単位世界は識別子を用いて区別される。

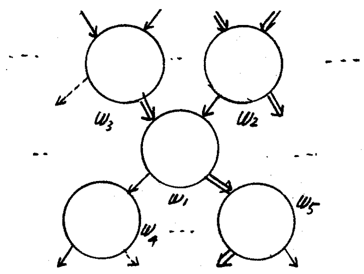


Fig1 多重世界

多重世界論理における論理式 wff は以下のように定義する。定義式中におけ記号のうち □,◇ は様相演算子であり、^,v は内包及び外延演算子である。それらについては後述する。

$$
\begin{aligned}
\langle wff \rangle ::= &\ \langle \text{等号を含む一階述語論理式} \rangle \\
&\mid \langle wff \rangle \wedge \langle wff \rangle \mid \langle wff \rangle \vee \langle wff \rangle \\
&\mid \sim \langle wff \rangle \mid \langle wff \rangle \supset \langle wff \rangle \\
&\mid \Box \langle wff \rangle \mid \Diamond \langle wff \rangle \\
&\mid \forall x \langle wff \rangle \mid \exists x \langle wff \rangle \\
&\mid {}^{\vee}{}^{\wedge} \langle wff \rangle
\end{aligned}
$$

2、2 renaming

内包及び外延演算子は異なる世界の間で述語や個体の名前を付けかえる操作のために導入した演算である。名前の付けかえは、従来の知識表現手法の中では重要視されていなかったが、我々はこの演算こそが知識表現において最も基本的かつ重要な演算であると考えている。多重世界論理による知識表現の多様性は、この renaming の概念によるところが非常に大である。

多重世界において、個々の世界での述語名、個体名等の表現は、それぞれの世界に対して固定されている。した

がって同一の対象を指示している表現が各世界で異なったり、同一な表現が世界により異なる対象を指示することが考えられる。例えば日本語、英語、独語の世界を考える。それぞれの世界での"本"，"book"，"Buch" という異なる表現は同一の概念（本）を指示している。

したがって注目する世界におけるある表現が、他の世界ではどんな表現に対応するかというrenamingなる概念を導入する必要がある。各世界でconstantに束縛された表現には述語名及び個体名等がある。それらが世界によりどのようにrenameされるかという点に注目する。このrenamingはoverloading，name conflict resolutionなどを実現するのに必要となる。

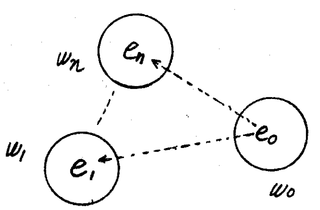


Fig2 renaming

Fig2は世界w_0における表現e_0が世界$w_1,\cdots,w_n$において表現$e_1,\cdots,e_n$に対応していることを表わしている。renamingを多重世界論理で実現するために内包なる概念を導入する。そのために多重世界論理では演算子として、ある表現をその内包に対応づける内包演算子$^\wedge$と、内包を外延である表現に対応づける外延演算子$^\vee$を導入する。世界w_0における表現e_0の内包は$^\wedge e_0$によって表わす。世界w_0における内包$^\wedge e_0$を次式で定義する。

$$
\begin{aligned}
{}^\wedge e_0 \triangleq \lambda w\,(cond\ & ((w=w_0)\ e_0) \\
& ((w=w_1)\ e_1) \\
& \quad \vdots \\
& ((w=w_n)\ e_n) \\
& (else\ \ \perp\ )) \quad \cdots ①
\end{aligned}
$$

ただし表現⊥はどの世界で評価してもunknownな表現とする。この表現が述語として表われたら、その述語⊥はどの世界で評価してもunknown、また個体項として表われたら、それを含む述語全体がunknownとなる。多重世界論理での内包とは、ある世界での表現が他の世界ではどの表現に対応するかを示す関数として定義する。$^\wedge e_0$では、どの世界での表現e_0の内包かがわからないので、新しい関数Intを次式のように定義する。

$$Int(e_0, w_0) \triangleq {}^\wedge e_0 \cdots (at\ w_0) \quad \cdots ②$$

次に定義された内包を、各世界のある表現に対応させる演算子$^\vee$（外延演算子）を導入する。具体的には内包$^\wedge e_0$に対し、ある世界w_iを適用させ、そこでの表現e_iに対応させる関数である。どの世界w_iが適用されるかは演算子$^\vee$がどの世界w_iで用いられたかで決まる。演算子$^\vee$では適用される世界を陽に示せないので、新しい関数Extを次式のように定義する。

$$
\begin{aligned}
{}^{\vee\wedge} e_0 &\triangleq Ext({}^\wedge e_0, w_i) \\
&= {}^\wedge e_0\,(w_i) \qquad \cdots ③
\end{aligned}
$$

①,②③式より

$$
\begin{aligned}
Ext(Int(e_0, w_0), w_i) &= Int(e_0, w_0)(w_i) \\
&= e_i \qquad \cdots ④
\end{aligned}
$$

いま世界w_0における表現e_0を世界w_iにおける対応する表現e_iにrenameする関数Ren_{w_0,w_i}を考える。この関数は④式より次のように定義できる。

$$Ren_{w_0,w_i}(e_0) \triangleq Ext(Int(e_0, w_0), w_i) \quad \cdots ⑤$$

一般にrenaming関数Renは

$$Ren = {}^{\vee\wedge} \quad \cdots ⑥$$

であると考えられる。

それぞれの世界では、そこで用いられる全ての表現に対する内包が定義さ

れていなければならない。その定義は各世界にある表(renaming table)に示されているものとする。実際には全ての表現に対する内包は示すことができないので、renaming table に内包が明記されていない表現e'は、どの世界でも同一な表現に対応するものとし、その内包${}^{\wedge}e'$を次式のように判断する。

$$ {}^{\wedge}e' = \lambda w\ e' \quad \cdots⑦ $$

いまn変項述語$P(x_1,\cdots,x_n)$に対するrenaming関数Renの適用を次式で定義する。

$$ Ren(P(x_1,\cdots,x_n)) \triangleq Ren(P)(Ren(x_1),\cdots,Ren(x_n)) \quad \cdots⑧ $$

また、$x_i\ (1 \leq i \leq n)$が個体変項であるなら

$$ Ren(x_i) \triangleq x_i \quad \cdots\cdots⑨ $$

と定義する。次に任意の多重世界論理式Aは、式Aに含まれる述語$(P_1,\cdots,P_n)$の真理値が定まれば、式全体としての真理値も定まる。その意味から、式Aとは述語$P_1,\cdots,P_n$を引数としてもつ真理値に関する関数とみることができる。したがって式Aは次式で表わされる関数である。いま右辺のφを式Aを示す文関数とする。

$$ A = \varphi(P_1,\cdots,P_n) \quad \cdots⑩ $$

いまrenaming関数Renを式Aに用いれるよう次式のように拡張する。

$$ Ren(A) \triangleq \varphi(Ren(P_1),\cdots,Ren(P_n)) \quad \cdots⑪ $$

⑪式を演算子$\vee$,$\wedge$を用いて表わせば⑥式より

$$ {}^{\vee\wedge}A \triangleq \varphi({}^{\vee\wedge}P_1,\cdots,{}^{\vee\wedge}P_n) \quad \cdots⑫ $$

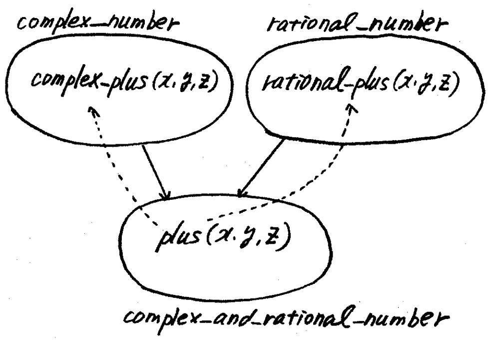
となる。具体的には任意の多重世界論理式A, Bに対し、

$$ {}^{\vee\wedge}(A \wedge B) \triangleq {}^{\vee\wedge}A \wedge {}^{\vee\wedge}B,\ {}^{\vee\wedge}(A \vee B) \triangleq {}^{\vee\wedge}A \vee {}^{\vee\wedge}B, $$
$$ {}^{\vee\wedge}(\sim A) \triangleq \sim {}^{\vee\wedge}A,\ {}^{\vee\wedge}(A \supset B) \triangleq {}^{\vee\wedge}A \supset {}^{\vee\wedge}B, $$
$$ {}^{\vee\wedge}\Box A \triangleq \Box {}^{\vee\wedge}A,\ {}^{\vee\wedge}\Diamond A \triangleq \Diamond {}^{\vee\wedge}A $$

と定義する。以上導入したrenamingなる概念を用い、overloading, name conflict resolution を表現する。いずれも多重継承に内在する本質的な問題点である。

例1 overloading

Fig3では世界complex_number及び世界rational_numberにおける述語complex-plus(x,y,z)とrational-plus(x,y,z)をそれらの子世界complex_and_rational_numberにおいて、plus(x,y,z)という一つの述語にoverloadしている例である。



complex-and-rational_number の renaming table

```
^plus = λw(cond ((w=complex_number)
                    complex-plus      )
                ((w=rational_number)
                    rational_plus     )
                (else     ⊥          ))
```

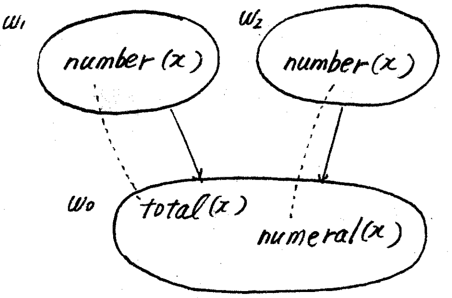
Fig3 overloading

overloading は renaming という点に注目すれば、親世界では異なる述語を子世界で同一述語に対し rename しているにすぎない。

例2 name conflict resolution

numberは総数(total)という意味と、数字(numeral)という意味がある。Fig4では、これらを混同しないように、

世界w_0では、その親世界での述語である number(x) を rename して、子世界では別々の述語として定義してある。



w_0のrenaming table

$$\begin{cases} \hat{}total = \lambda w\,(cond\,((w = w_1)\ number) \\ \qquad\qquad\qquad (else \quad \bot)) \\ \hat{}numeral = \lambda w\,(cond\,((w = w_2)\ number) \\ \qquad\qquad\qquad (else \quad \bot)) \end{cases}$$

Fig 4　name conflict resolution

name conflict resolutionは、ある世界では指示する対象が異なるが同一表現だった述語を別世界で別表現の述語にrenamingして区別することで解消する。

2.3 様相演算子（□，◇）

多重世界論理式の定義の中に出てきた記号□，◇は親世界における証明可能性を示す様相演算子である。それらの定義は以下の通りである。ただしAは任意の多重世界論理式，Wは世界wの親世界$w_1,\ldots,w_n$の集合、$\vdash_w P$ はPが世界wにおいて証明可能であることを示す。

□A："□A"が記述されている世界(w)の親世界全てで、式Aに対応する各世界でのrenameされた式が証明可能

$$\begin{cases} \forall w_i \in W,\ \vdash_{w_i} Ren_{w,w_i}(A) \Rightarrow \vdash_w \Box A \\ otherwise \quad \Rightarrow \vdash_w \sim\Box A \end{cases} \quad \cdots ⑬$$

◇A："◇A"が記述されている世界(w)の親世界の少なくとも一つ世界で、式Aをその世界でrenameした式が証明可能

$$\begin{cases} \exists w_i \in W,\ \vdash_{w_i} Ren_{w,w_i}(A) \Rightarrow \vdash_w \Diamond A \\ otherwise \quad \Rightarrow \vdash_w \sim\Diamond A \end{cases} \quad \cdots ⑭$$

様相演算子□，◇はどちらも、証明できなければfalseと判定する。いいかえれば様相演算子□，◇は閉世界仮説を採用している。

2.4 世界間リンク

世界間リンクが行なうことは、次の3つのことである。

① 子世界に記述されている様相演算子の作用範囲にある親世界を示す。
② 子世界からvisible（世界内に記述された式の集合が参照できる）な親世界を示す。
③ 子世界で証明できない述語について推論するために用いられる親世界を示す。

世界間リンクには①のみを行なうreferリンクと、①,②を行なうcopyリンクと、①,③を行なうinheritanceリンクがある。これらのリンクを形式的に扱うためにrenaming関数Renを拡張する。Sを式$A_1,\ldots,A_n$の集合（$\{A_1,\ldots,A_n\}$）とする。このとき式の集合Sに対するrenaming関数Renの適用を次式で定義する。

$$Ren(S) \triangleq \{Ren(A_1),\ldots,Ren(A_n)\} \quad \cdots ⑮$$

2.4.1 referリンク（$\dashrightarrow$）

referリンクは2.3で説明した様相演算子の作用範囲にある親世界を示すこ

とのみを行なう。

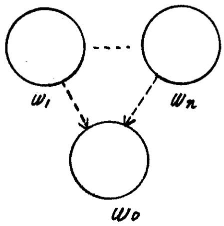


Fig5. refer リンク

Fig5のように世界w_0と世界$w_1,\cdots,w_n$間にreferリンクが張られたとする。このとき世界w_0における様相演算子□,◇の証明可能性を調べる親世界が$w_1,\cdots,w_n$になる。

2、4、2 copyリンク(⇒)

copyリンクは、このリンクにより関連づけられた親世界で利用可能な式全てをrenameして子世界へ持ってくる。

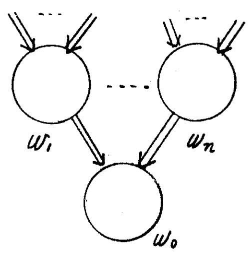


Fig6. copyリンク

Fig6のように、世界w_0がcopyリンクにより世界$w_1,\cdots,w_n$と関連づけられているとする。いま世界w_0に記述された式の集合をS_{w_0}とする。また世界w_0で利用可能な式の集合を$\overline{S}_{w_0}$と表現すると$\overline{S}_{w_0}$は次式で定義される。

$$\overline{S}_{w_0} \triangleq S_{w_0} \cup Ren_{w_1,w_0}(\overline{S}_{w_1}) \cup \cdots \cdots \cup Ren_{w_n w_0}(\overline{S}_{w_n}) \quad \cdots\cdots (16)$$

ただし、任意の世界w_iについて、世界w_iがcopyリンクについての子世界になっていなければ、

$$\overline{S}_{w_i} \triangleq S_{w_i} \quad \cdots (17)$$

と定義する。任意の世界wについて

$$\overline{S}_w \vdash P \Rightarrow \underset{w}{\vdash} P \quad \cdots (18)$$

が成立するとする。任意の世界wについて、世界wの利用可能な式の集合$\overline{S}_w$から証明できるならば、世界wで証明できると考える。

2、4、3 inheritanceリンク(⟶)

inheritanceリンクでは注目した子世界だけでは証明不可能な述語は、子世界とinheritanceリンクで関連づけられた親世界での利用可能な式の集合を合わせ、推論する。それでも証明できない場合、選んだ親世界に対するinheritanceリンクについての親世界の一つを選び、そこの利用可能な式の集合も合わせて推論を行なう。このように証明できるまでinheritanceリンクを逆にたどっていく。

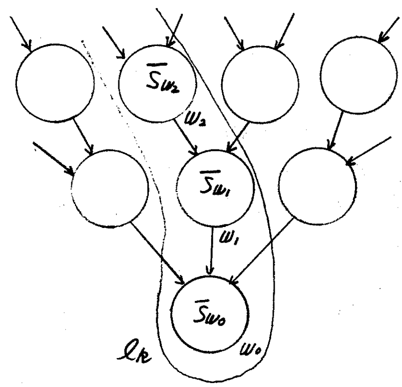


Fig7 inheritanceリンク

Fig7において、世界w_0からinheritanceリンクを逆にたどり、それ以上inheritanceリンクをたどれなくなるまでたどっていくとき表われる全ての世界の列(例えばFig5での$w_0, w_1, w_2, \cdots, w_n$)を、世界$w_0$の探索路と呼ぶ。世界$w_0$のある探索路$l_k(w_0, w_1, \cdots, w_n)$に対し、$\hat{S}_{w_1}, \cdots, \hat{S}_{w_n}$及び$P_0, P_1, \cdots, P_n$を次のように定義する。

$$\begin{cases} \hat{S}_{w_0} \triangleq \bar{S}_{w_0} \\ \hat{S}_{w_1} \triangleq \bar{S}_{w_1} \cup Ren_{w_0,w_1}(\hat{S}_{w_0}) \\ \vdots \\ \hat{S}_{w_i} \triangleq \bar{S}_{w_i} \cup Ren_{w_{i-1},w_i}(\hat{S}_{w_{i-1}}) \\ \vdots \end{cases} \quad (19)$$

$$\begin{cases} P_1 \triangleq Ren_{w_0,w_1}(P_0) \\ P_2 \triangleq Ren_{w_1,w_2}(P_1) \\ \vdots \\ P_i \triangleq Ren_{w_{i-1},w_i}(P_{i-1}) \\ \vdots \end{cases} \quad (20)$$

$\hat{S}_{w_i}$は世界$w_0,\dots,w_i$で利用可能な式の集合$\bar{S}_{w_0},\dots,\bar{S}_{w_{i-1}}$を探索路$l_k$に沿って、renameしながら順次、和集合を作り、最後に$\bar{S}_{w_i}$との和をとった集合である。P_iは世界w_0での述語P_0が探索路に沿ってrenamingを繰り返した結果、対応する述語である。

　世界w_0における利用可能な式の集合$\bar{S}_{w_0}$からP_0及び$\sim P_0$が証明できないとき、式で書けば次式のとき

$$\bar{S}_{w_0} \nvdash P_0 \text{ and } \bar{S}_{w_0} \nvdash \sim P_0 \quad (21)$$

次のように推論する。世界w_0のある探索路$l_k(w_0, w_1, \dots, w_n)$を考える。このとき$1 \le i \le n$, $1 \le j < i$なるi, jに対して

$$\forall w_j\ \hat{S}_{w_j} \nvdash P_j,\ \forall w_j\ \hat{S}_{w_j} \nvdash \sim P_j$$

のとき

$$\begin{cases} \hat{S}_{w_i} \vdash P_i \Longrightarrow \underset{l_k}{\Vdash} P_0 \\ \hat{S}_{w_i} \vdash \sim P_i \Longrightarrow \underset{l_k}{\Vdash} \sim P_0 \\ \hat{S}_{w_i} \nvdash P_i,\ \hat{S}_{w_i} \nvdash \sim P_i \Longrightarrow \underset{l_k}{\nVdash} P_0,\ \underset{l_k}{\nVdash} \sim P_0 \end{cases}$$

ただし$\underset{l_k}{\Vdash} P$は探索路l_k上では最初にPに対応する述語が証明可能であることを示す。全ての探索路について同様に推論した結果により次のように考える。いま世界w_0の全ての探索路の集合をLで表わす。

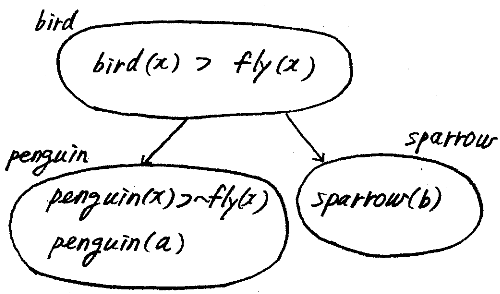
$$\begin{cases} \exists l_k \in L, \underset{l_k}{\Vdash} P_0,\ \forall l_k \in L, \underset{l_k}{\nVdash} \sim P_0 \Rightarrow \underset{w_0}{\vdash} P_0,\ \underset{w_0}{\nVdash} \sim P_0 \\ \qquad (\text{true}) \\ \exists l_k \in L, \underset{l_k}{\Vdash} \sim P_0,\ \forall l_k \in L, \underset{l_k}{\nVdash} P_0 \Rightarrow \underset{w_0}{\vdash} \sim P_0,\ \underset{w_0}{\nVdash} P_0 \\ \qquad (\text{false}) \\ \exists l_k \in L, \underset{l_k}{\Vdash} P_0,\ \exists l_m \in L, \underset{l_m}{\Vdash} \sim P_0 \Rightarrow \underset{w_0}{\vdash} P_0,\ \underset{w_0}{\vdash} \sim P_0 \\ \qquad (\text{矛盾}) \\ \forall l_k \in L, \underset{l_k}{\nVdash} P_0,\ \forall l_k \in L, \underset{l_k}{\nVdash} \sim P_0 \Rightarrow \underset{w_0}{\nVdash} P_0,\ \underset{w_0}{\nVdash} \sim P_0 \\ \qquad (\text{unknown}) \end{cases}$$

3、具体例

　次に多重世界論理を用いて、デフォールトリーズニング、サーカムスクリプション、親世界の知識間に矛盾が存在する場合の多重継承についての実現方法を示す。

3.1 デフォールトリーズニング

　デフォールトリーズニングの例として、"通常鳥は飛ぶが、ペンギンは飛ばない"という知識を表現する。



世界penguinのrenaming table

$$\begin{cases} \hat{}penguin = \lambda w\,(cond\,((w = bird)\ bird) \\ \qquad\qquad (else\ \ \bot)) \end{cases}$$

世界sparrowのrenaming table

$$\begin{cases} \hat{}sparrow = \lambda w\,(cond\,((w = bird)\ bird) \\ \qquad\qquad (else\ \ \bot)) \end{cases}$$

Fig 8 default reasoning

いまFig8で表わされるように世界penguin, sparrow, birdが配置されているとする。世界penguin, sparrowに対し、質問fly(a)?, fly(b)? を行なうことを考える。以下、式中では世界penguin, sparrow, birdをp, s, bと略し、S_p, S_s, S_bはそれぞれの世界に記述された式の集合である。

$$S_p \vdash \sim fly(a)$$

が成立する。したがって世界penguinへの質問fly(a)?の答えはfalseである。これ以上推論されない。世界sparrowに対する質問fly(b)?は

$$S_s \nvdash fly(b) \ , \ S_s \nvdash \sim fly(b)$$

よってinheritanceリンクを用い、親世界birdの知識を合わせて推論する。

$$Ren_{s,b}(\hat{S_s}) = Ren_{s,b}(S_s) = \{bird(b)\}$$

$$Ren_{s,b}(fly(b)) = fly(b)$$

$$\bar{S_b} = S_b \ , \ \hat{S_b} = S_b \cup Ren_{s,b}(S_s)$$

$$\hat{S_b} = \{bird(b), bird(x) \supset fly(x)\}$$

$$\therefore \hat{S_b} \vdash Ren_{s,b}(fly(b))$$

$$\therefore \vdash_{s} fly(b)$$

よって世界sparrowに対する質問fly(b)?の答えはtrueであると推論することができる。

3、2 サーカムスクリプション

サーカムスクリプションとはある述語について証明できるもののみが真、他を偽と判断する方法である。

Fig9において世界w_1を積木の世界とする。世界w_0には述語blockに関する式は図に示した式しかないとする。このとき世界w_0では述語blockは世界w_1での述語blockについてサーカムスクリプションをかけたものになる。

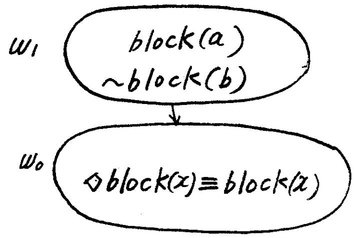


Fig 9 circumscription

世界w_0に対し、質問block(a)? 及びblock(b)?, block(c)? を行なう。このとき

$$\vdash_{w_1} block(a) \Rightarrow \vdash_{w_0} \Diamond block(a)$$

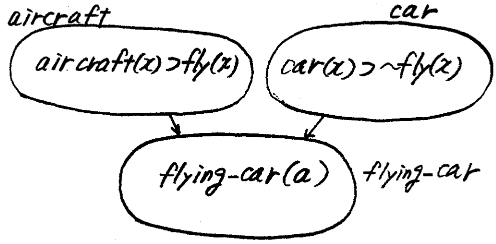
$$\nvdash_{w_1} block(b) \Rightarrow \vdash_{w_0} \sim\Diamond block(b)$$

$$\nvdash_{w_1} block(c) \Rightarrow \vdash_{w_0} \sim\Diamond block(c)$$

よって ◇block(a), ~◇block(b), ~◇block(c) 及び ◇block(x)≡block(x) より、block(a), ~block(b), ~block(c) が証明される。3つの質問に対する世界w_0での答えはtrue, false, falseの順になる。したがって世界w_0において積木である個体は、世界w_1で積木であると宣言されたものに限定される。

3、3 親世界の知識間に矛盾が存在する場合

世界flying_carは世界aircraft及びcarより多重継承する(Fig 10)。このとき述語flyに関して、両方の親世界の知識をそのまま継承すると矛盾が生じる。



(Fig 10, 次ページに続く)

世界flying_carのrenaming table

$$
\left\{
\begin{array}{l}
{}^{\wedge}flying\_car \\
\quad = \lambda w\,(cond\,((w{=}car)\ car\ ) \\
\qquad\qquad ((w{=}aircraft)\ aircraft\ ) \\
\qquad\qquad (else\ \ \perp\ \ )) \\
{}^{\wedge}fly\ \ = \lambda w\,(cond\,((w{=}aircraft)\,fly\ \ ) \\
\qquad\qquad (else\ \ \ \perp\ )) \\
\end{array}
\right.
$$

Fig 10 親世界の知識間に矛盾のある多重継承

flyに関する知識は世界aircraftのものを継承するように述語fly の内包^fly を定義してある。世界flying_carに対し質問fly(a)?を与えた場合の推論を以下に示す。ただし式中で世界flying_car, aircraft, carをf, a, cと略し、またS_f, S_a, S_cをそれぞれの世界での式の集合とする。

$$S_f \not\vdash fly(a) \text{ and } S_f \not\vdash \sim fly(a)$$

より世界flying_carのみではunknownであるから、親世界の知識も用いて推論する。親世界carを用い推論すると、

$$S_c = \{ car(x) \supset \sim fly(x) \}$$

$$Ren_{f,c}(S_f) = \{ car(a) \}$$

$$Ren_{f,c}(fly(a)) = \perp$$

よって

$$S_c \cup Ren_{f,c}(S_f) \not\vdash Ren_{f,c}(fly(a))$$

$$\text{〃} \qquad \not\vdash \sim Ren_{f,c}(fly(a))$$

したがって世界carの知識を用いても世界flying_carでfly(a)が証明できない。世界aircraftの知識を用いた場合、

$$S_a = \{ aircraft(x) \supset fly(x) \}$$

$$Ren_{f,a}(S_f) = \{ aircraft(a) \}$$

$$Ren_{f,a}(fly(a)) = fly(a)$$

よって

$$S_a \cup Ren_{f,p}(S_f) \vdash Ren_{f,a}(fly(a))$$

は成立する。したがって世界flying_carに対する質問fly(a)の答えはtrueである。よって矛盾せずに継承される。この例により知識の選択的な継承がrenamingを用いることで可能であることがわかる。

4、結論

知識表現を多重世界モデルを用い、統一的かつ形式的に取り扱う試みとして、証明可能性を示す様相演算子及びrenamingを行なう演算子、知識の継承方法等を規定する世界間リンクを導入した多重世界論理を示した。しかし現在の時点ではPART_OF, update, 時間変化の表現、多重世界から情報を引き出す効率のよい方法などが未解決であり、今後の課題となっている。

5、参考文献

[1] Reiter, R.: "A logic for default reasoning", ibid., PP.81–132

[2] 松本裕治: "Default Reasoningと非単調論理" 電子通信学会誌, 64, 3, PP.313–316

[3] McCarthy, J.: "Circumscription – A Form of Non-Monotonic Reasoning", ibid., PP.27–39

[4] Clark, K.: "Negation as failure", "Logic and Data Bases", Plenum Press, PP.293–322

[5] 内田種臣: "様相の論理", 早稲田大学出版部